Part I. Generating a single multinomial sample given an input sample size m and a probability vector $(\theta_1, ..., \theta_k)$.

```
# function that generates N multinomial samples given an input sample size m and a probability vector
multinom.generator <- function(N, m, t) {
 k <- length(t)
 x <- matrix(data=0, nrow = N, ncol = k, byrow = T)
 x <- data.frame(x)
 s <- rbinom(N, size=m, prob=t[1])
 x[1:N, 1] <- s
 sum.t <- 1-cumsum(t)
 sum.x <- matrix(data=0, nrow = N, ncol = k, byrow = T)
 sum.x[1:N, 1] <- x[1:N, 1]
        if(k > 2)
        {
                for (i in 2:(k-1))
                {
                        x[1:N, i] <- rbinom(N, size=m-sum.x[1:N,(i-1)], prob=t[i]/sum.t[i-1])
                        sum.x <- t(apply(x, 1, cumsum))

                         for(j in 1:N)
                         {
                                if (sum.x[j,i-1]==m)
                                {
                                x[j, i] <- 0
                                return(x)
                        }
                 }
            }
         }
 x[1:N, k] <- rbinom(N, size=m-sum.x[1:N, k-1], prob=1)
 return(x)
 }
# generate 2 samples, m=100, prob=c(0.1,0.2,0.3, 0.4)
multinom.generator(2, 100, c(0.1,0.2,0.3, 0.4))


##   X1 X2 X3 X4
## 1 13 21 26 40
## 2  8 18 30 44
```

```
library(reshape2)
library(ggplot2)
multinom.sample <-  multinom.generator(100, 100, c(0.1,0.2,0.3, 0.4))

multinom.sample <- suppressWarnings(melt(data.frame(multinom.sample))) # reshape data


## No id variables; using all as measure variables


p <- ggplot(multinom.sample, aes(x = variable, y = value))
```

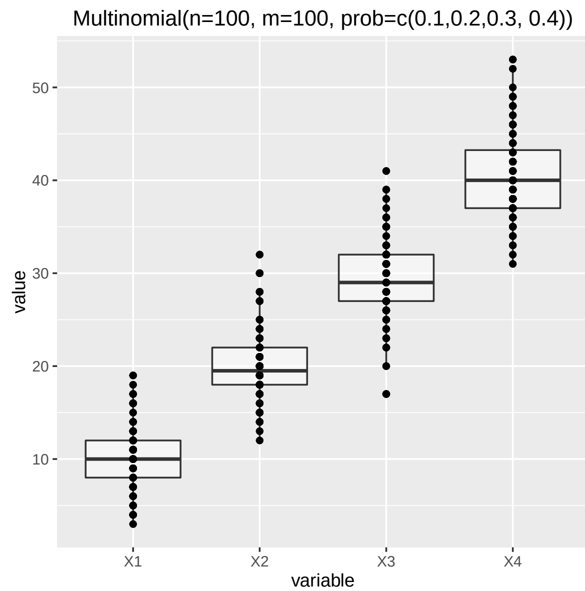

Figure 1: Boxplots of 100 multinomial samples with sample size equals 100.

```
p <- p + geom_boxplot(alpha = 0.5)
p <- p + geom_point()
p <- p + labs(title = "Multinomial(n=100, m=100, prob=c(0.1,0.2,0.3, 0.4))")
print(p)
```

Part II. Multinomial hypothesis testing.

```
# computes the Pearson statistic
pearson.stat <- function(x,theta){
  m <- sum(x)
  sum((x - m*theta)^2/(m*theta))
}

# computes the likelihood ratio statistic
lrt.stat <- function(x,theta){
  m <- sum(x)
  sum(2*x*log((x+1e-10)/(m*theta)))
}
```

```
# check that input values are 'permissible'
theta.check <- function(theta){
   if(any(theta) < 0 | any(theta) > 1)
    {
      stop("The probabilities should be between 0 and 1")
    }

    if(length(theta) < 2)
```

```r
    {
      stop("The probability vector should have at least length 2")
    }

    if(abs(sum(theta) - 1) > 1e-10)
    {
      stop("The probabilities should sum to 1")
    }
  return(theta)
}

# check x
arg.check <- function(x){
    if(!all(x) > 0 & !all(x) == 0)
    {
      stop("The counts should be non-negative")
    }

    if(length(x) < 2)
    {
      stop("The x vector should have at least length 2")
    }
   return(x)
}

# function that checks inputs and return statistic values with their p-values
 stat.comp <- function(x, theta){
    suppressWarnings(arg.check(x))
    suppressWarnings(theta.check(theta))
    p.st <- pearson.stat(x,theta)
    lrt.st <- lrt.stat(x,theta)
    pv.pearson <- 1 - pchisq(p.st, df = length(theta) - 1)
    pv.lrt <- 1 - pchisq(lrt.st, df = length(theta) - 1)
    cat("Pearson's test", "\n", "Statistic:",  p.st, "\n", "P-value:", pv.pearson,"\n")
    cat("\n")
    cat("The likelihood ratio test", "\n", "Statistic:",  lrt.st, "\n", "P-value:", pv.lrt,"\n")
}

# usage
x <- c(16, 50, 31, 11, 9)
theta <- c(0.1, 0.2, 0.4, 0.2, 0.1)
stat.comp(x, theta)


## Pearson's test
##  Statistic: 44.34615
##  P-value: 5.436742e-09
##
## The likelihood ratio test
##  Statistic: 39.07809
##  P-value: 6.712448e-08
```

Lets generalize P and G^2 functions so that they can compute each statistic for multiple samples, given by rows of a matrix.

```
pearson.stat <- function(n, x, theta) {
  p<-rep(0, n)
  pv<-rep(0, n)
  mx <- rep(0, length(x))
  mx <- matrix(x, nrow = n, byrow = TRUE)
  rs <- rowSums(mx)
  cat("Pearson's test", "\n")
  for (i in 1:n) {
    m <- rs[i]
    t<-0
    for (j in 1:length(theta)) {
      t <- t + (mx[i,j] - m*theta[j])^2/(m*theta[j])
    }
    p[i] <- t
    pv[i] <- 1 - pchisq(p[i], df = length(theta) - 1)
    cat("\n")
    cat(i, "Statistic:",  p[i], "\n", "P-value:", pv[i],"\n")
  }
}

lrt.stat <- function(n, x, theta) {
  p<-rep(0, n)
  pv<-rep(0, n)
  mx <- rep(0, length(x))
  mx <- matrix(x, nrow = n, byrow = TRUE)
  rs <- rowSums(mx)
  cat("The likelihood ratio test", "\n")
  for (i in 1:n) {
    m <- rs[i]
    t<-0
    for (j in 1:length(theta)) {
      t <- t + 2*mx[i,j]*log((mx[i,j]+1e-10)/(m*theta[j]))
    }
    p[i] <- t
    pv[i] <- 1 - pchisq(p[i], df = length(theta) - 1)
    cat("\n")
    cat(i, "Statistic:",  p[i], "\n", "P-value:", pv[i],"\n")
  }
}
```

```
# illustration on the given data
x <- c(16, 50, 31, 11, 9,
       10, 23, 22, 20, 7,
       21, 10, 42, 3, 1,
       3, 12, 31, 16, 0)
theta <- c(0.1, 0.2, 0.4, 0.2, 0.1)

pearson.stat(4, x, theta)
```

```
## Pearson's test
##
## 1 Statistic: 44.34615
##  P-value: 5.436742e-09
##
## 2 Statistic: 7.573171
##  P-value: 0.1085257
##
## 3 Statistic: 44.75325
##  P-value: 4.47443e-09
##
## 4 Statistic: 10.45968
##  P-value: 0.03335698
```

```
lrt.stat(4, x, theta)
```

```
## The likelihood ratio test
##
## 1 Statistic: 39.07809
##  P-value: 6.712448e-08
##
## 2 Statistic: 7.676726
##  P-value: 0.1041642
##
## 3 Statistic: 45.65908
##  P-value: 2.899826e-09
##
## 4 Statistic: 16.84887
##  P-value: 0.002068087
```